applied sciences

MDPI

Article

Single Image Super-Resolution Method Using CNN-Based Lightweight Neural Networks

Seonjae Kim [1], Dongsan Jun [2,*], Byung-Gyu Kim [3], Hunjoo Lee [4] and Eunjun Rhee [4]

[1] Department of Convergence IT Engineering, Kyungnam University, Changwon 51767, Korea; sjkim@kyungnam-ispl.kr
[2] Department of Information and Communication Engineering, Kyungnam University, Changwon 51767, Korea
[3] Department of IT Engineering, Sookmyung Women's University, Seoul 04310, Korea; bg.kim@sookmyung.ac.kr
[4] Intelligent Convergence Research Lab., Electronics and Telecommunications Research Institute (ETRI), Daejeon 34129, Korea; hjoo@etri.re.kr (H.L.); ejrhee@etri.re.kr (E.R.)
* Correspondence: dsjun9643@kyungnam.ac.kr

Abstract: There are many studies that seek to enhance a low resolution image to a high resolution image in the area of super-resolution. As deep learning technologies have recently shown impressive results on the image interpolation and restoration field, recent studies are focusing on convolutional neural network (CNN)-based super-resolution schemes to surpass the conventional pixel-wise interpolation methods. In this paper, we propose two lightweight neural networks with a hybrid residual and dense connection structure to improve the super-resolution performance. In order to design the proposed networks, we extracted training images from the DIVerse 2K (DIV2K) image dataset and investigated the trade-off between the quality enhancement performance and network complexity under the proposed methods. The experimental results show that the proposed methods can significantly reduce both the inference speed and the memory required to store parameters and intermediate feature maps, while maintaining similar image quality compared to the previous methods.

Keywords: deep learning; convolutional neural networks; lightweight neural network; single image super-resolution; image enhancement; image restoration; residual dense networks

Citation: Kim, S.; Jun, D.; Kim, B.-G.; Lee, H.; Rhee, E. Single Image Super-Resolution Method Using CNN-Based Lightweight Neural Networks. *Appl. Sci.* **2021**, *11*, 1092. https://doi.org/10.3390/app11031092

Received: 10 December 2020
Accepted: 22 January 2021
Published: 25 January 2021

1. Introduction

While the resolution of images has been rapidly increasing in recent years with the development of high-performance cameras, advanced image compression, and display panels, the demands to generate high resolution images from pre-existing low-resolution images are also increasing for rendering on high resolution displays. In the field of computer vision, single image super-resolution (SISR) methods aim at recovering a high-resolution image from a single low-resolution image. Since the low-resolution images cannot represent the high-frequency information properly, most super-resolution (SR) methods have focused on restoring high-frequency components. For this reason, SR methods are used to restore the high-frequency components from quantized images at the image and video post-processing stage [1–3].

Deep learning schemes such as convolutional neural network (CNN) and multi-layer perceptron (MLP) are a branch of machine learning which aims to learn the correlations between input and output data. In general, the output in the process of the convolution operations is one pixel, which is a weighted sum between an input image block and a filter, so an output image represents the spatial correlation of input image corresponding to the filters used. As CNN-based deep learning technologies have recently shown impressive results in the area of SISR, various CNN-based SR methods have been developed that surpass the conventional SR methods, such as image statistical methods and patch- based methods [4,5]. In order to improve the quality of low-resolution images, CNN-based SR networks tend to deploy more complicated schemes, which have deeper and denser CNN structures and cause increases in the computational complexity like the required memory to store network parameters, the number of convolution operations, and the inference speed. We propose two SR-based lightweight neural networks (LNNs) with hybrid residual and dense networks, which are the "inter-layered SR-LNN" and "simplified SR-LNN", respectively, which we denote in this paper as "SR-ILLNN" and "SR-SLNN", respectively. The proposed methods were designed to produce similar image quality while reducing the number of networks parameters, compared to previous methods. Those SR technologies can be applied to the pre-processing stages of face and gesture recognition [6–8].

The remainder of this paper is organized as follows: In Section 2, we review previous studies related to CNN-based SISR methods. In Section 3, we describe the frameworks of the proposed two SR-LNNs for SISR. Finally, experimental results and conclusions are given in Sections 4 and 5, respectively.

2. Related Works

Deep learning-based SR methods have shown high potential in the field of image interpolation and restoration, compared to the conventional pixel-wise interpolation algorithms. Dong et al. proposed a three-layer CNN structure called super-resolution convolutional neural network (SR-CNN) [9], which learns an end-to-end mapping from a bi-cubic interpolated low-resolution image to a high-resolution image. Since the advent of SR-CNN, a variety of CNN networks with deeper and denser network structure [10–13] have been developed to improve the accuracy of SR.

In particular, He et al. proposed a ResNet [11] for image classification. Its key idea is to learn residuals through global or local skip connection. It notes that ResNet can provide a high-speed training process and prevent the gradient vanishing effects. In addition to ResNet, Huang et al. proposed densely connected convolutional networks (DenseNet) [12] to combine hierarchical feature maps available along the network depth for more flexible and richer feature representations. Dong et al. proposed an artifacts reduction CNN (AR-CNN) [14], which effectively reduces the various compression artifacts such as block artifacts and ringing artifacts on Joint Photographic Experts Group (JPEG) compression images.

Kim et al. proposed a super-resolution scheme with very deep convolutional networks (VDSR) [15], which is connected with 20 convolutional layers and a global skip connection. In particular, the importance of receptive field size and the residual learning was verified by VDSR. Leding et al. proposed a SR-ResNet [16], which was designed with multiple residual blocks and generative adversarial network (GAN) for improving visually subjective quality. Here, a residual block is composed of multiple convolution layers, a batch normalization, and a local skip connection. Lim et al. exploited enhanced deep super-resolution (EDSR) and multi-scale deep super-resolution (MDSR) [17]. In particular, as these networks have been modified in a way of removing the batch normalization, it can reduce graphics processing unit (GPU) memory demand by about 40% compared with SR-ResNet.

Tong et al. proposed an image super-resolution using dense skip connections (SR-DenseNet) [18] as shown in Figure 1. Because SR-DenseNet consists of eight dense blocks and each dense block contains eight dense layers, this network has a total of 67 convolution layers and two deconvolution layers. Because the feature maps of the previous convolutional layer are concatenated with those of the current convolutional layer within a dense block, total number of the feature map from the last dense block reaches up to 1040 and it requires more memory capacity to store the massive network parameters and intermediate feature maps.

On the other hand, the aforementioned deep learning-based SR methods are also applied to compress raw video data. For example, Joint Video Experts Team (JVET) formed the Adhoc Group (AhG) for deep neural networks based video coding (DNNVC) [19] in 2020, which aims at exploring the coding efficiency using the deep learning schemes. Several studies [20–22] have shown better coding performance than the-state-of-the-art video coding technologies.

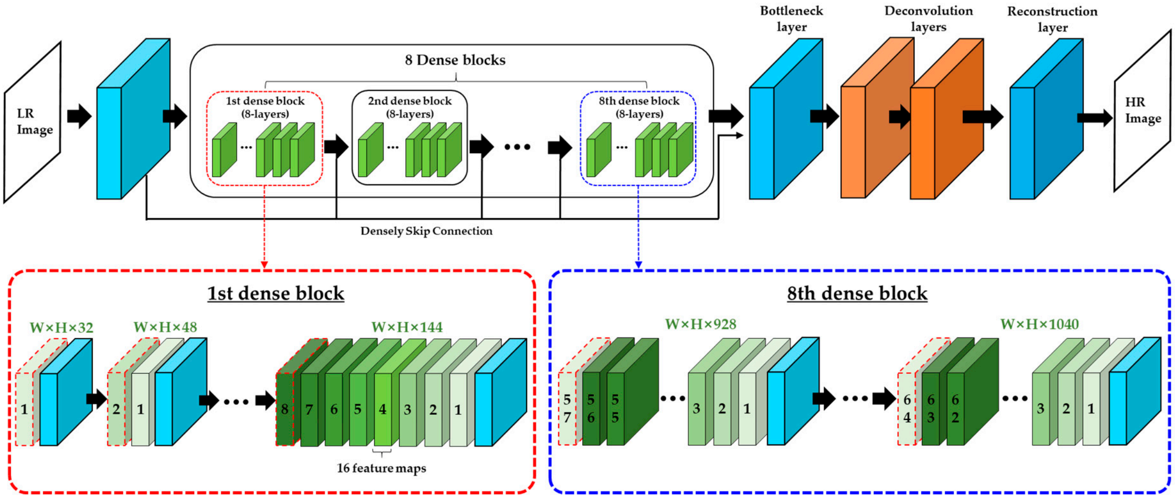


Figure 1. The framework of SR-DenseNet [18].

3. Proposed Method

Although more complicated deep neural network models have demonstrated better SR performance than conventional methods, it is difficult to implement them on low-complexity, low-power, and low-memory devices, due to the massive network parameters and convolution operations of deeper and denser networks. In case of SR-DenseNet, it is difficult to implement this model to the applications for real-time processing even though its SR performance is superior to that of other neural network models. To address this issue, we considered two lightweight network structures at the expense of unnoticeable quality degradation, compared to SR-DenseNet. The purpose of the proposed two lightweight neural networks for SISR is to quadruple the input images the same as SR-DenseNet. Firstly, SR-ILLNN learns the feature maps, which are derived from both low-resolution and interpolated low-resolution images. Secondly, SR-SLNN is designed to use only low-resolution feature maps of the SR-ILLNN for a few more reducing the network complexity.

3.1. Architecture of SR-ILLNN

Figure 2 shows the proposed SR-ILLNN, which consists of two inputs, 15 convolution layers and two deconvolution layers. The two inputs are denoted as a low-resolution (LR) image X_{LR} and a bi-cubic interpolated low-resolution (ILR) image X_{ILR} where N and M denote the width and height of the input image X_{LR}, respectively. The reason why we deployed the two inputs is to compensate the dense LR features of SR-DenseNet with high-resolution (HR) features of X_{ILR}, while reducing the number of convolutional layers as many as possible. As depicted in Figure 2, it consists of three parts, which are LR feature layers from convolutional layer 1 (Conv1) to Conv8, HR feature layers from Conv9 to Conv12, and shared feature layers from Conv13 to Conv15.

Each convolution is operated as in (1), where W_i, B_i, and '$\otimes$' represent the kernels, biases, and convolution operation of the *i*th layer, respectively. In this paper, we notate a kernel as $[F_w \times F_h \times F_c]$, where $F_w \times F_h$ and F_c are the spatial size of filter and the number of channels, respectively:

$$F_i(X_{LR}) = max(0, W_i \otimes F_{i-1}(X_{LR}) + B_i), \quad (1)$$

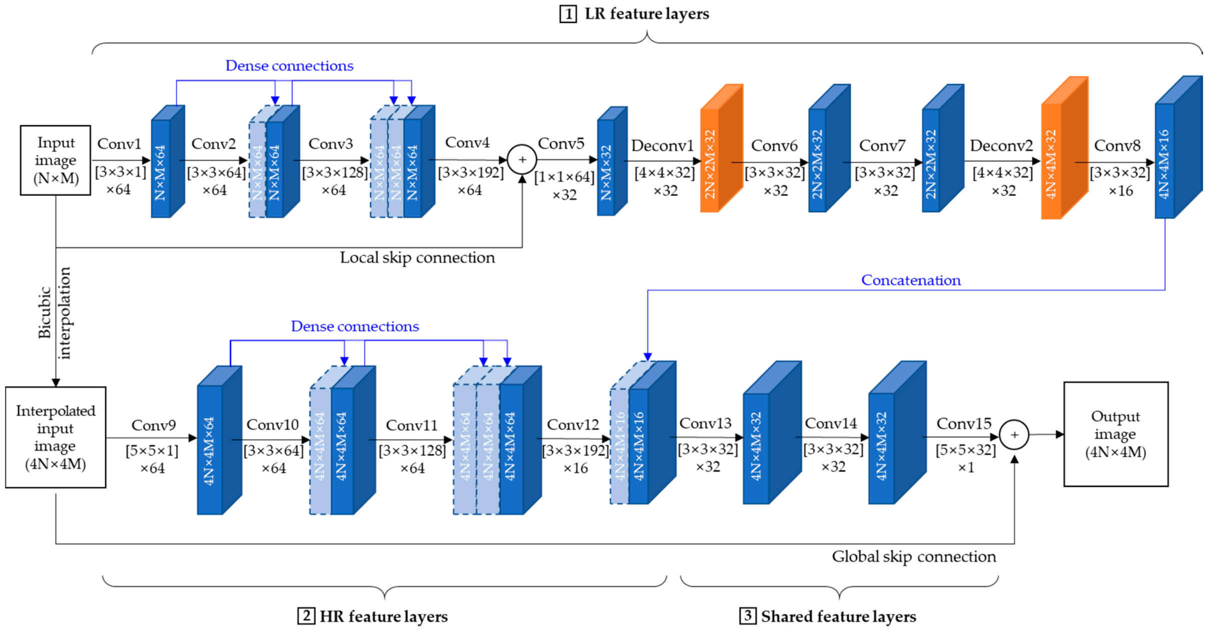


Figure 2. The framework of the proposed SR-ILLNN.

In the process of convolution operation, we applied rectified linear unit (ReLU, $max(0,\ x)$) on the filter responses and used a partial convolution-based padding scheme [23] to avoid the loss of boundary information. The padding sizes is defined so that the feature maps between different convolution layers can have the same spatial resolution as follows:

$$Padding\ Size\ =\ Floor(F_w/2), \tag{2}$$

where Floor(x) means the rounding down operation. Note that the convolutional layers of Conv1–4 and Conv9–12 of Figure 2 are conducted to generate output feature maps with dense connections for more flexible and richer feature representations, which are concatenated the feature maps of the previous layer with those of the current layer. So, convolution operations with dense connections are calculated as in (3):

$$\begin{aligned} F_i(X_{LR})\ &=\ max(0,\ W_i\ \otimes\ [F_1(X_{LR}),\ \ldots,\ F_{i-1}(X_{LR})]\ +\ B_i), \\ F_j(X_{ILR})\ &=\ max\left(0,\ W_j\ \otimes\ \left[F_9(X_{ILR}),\ \ldots,\ F_{j-1}(X_{ILR})\right]\ +\ B_j\right) \end{aligned} \tag{3}$$

A ResNet scheme [11] with skip connections can provide a high-speed training and prevent the gradient vanishing effect, so we deployed a local and a global skip connection to train the residual at the output feature maps of Conv4 and Conv15. Because the output feature maps F_4 and X_{LR} have the different number of channels in local skip connection, X_{LR} is copied up to the number of channels of F_4 before operating Conv5.

It should be noted that the number of feature maps has a strong effect on the inference speed. Therefore, the proposed SR-LNNs is designed to reduce the number of feature maps from 192 to 32, before deconvolution operation. Then, Deconv1 and Deconv2 are operated for image up-sampling as follows:

$$F_{deconv}(X_{LR})\ =\ max(0,\ W_{deconv}\ \odot\ F_{i-1}(X_{LR})\ +\ B_{deconv}), \tag{4}$$

where W_{deconv}, B_{deconv} are the kernels and biases of the deconvolution layer, respectively, and the symbol '$\odot$' denotes the deconvolution operation. As each deconvolution layer has different kernel weights and biases, it is superior to the conventional SR methods like pixel-wise interpolation methods.

In the stage of the shared feature layers, the output feature maps of the LR feature layers $F_8(X_{LR})$ are concatenated with those of HR feature layers $F_{12}(X_{ILR})$. Then, the concatenated feature maps $[F_8(X_{LR}),\ F_{12}(X_{ILR})]$ are inputted to the shared feature layers as in (5):

$$F_{13}(X) = max(0,\ W_{13} \otimes [F_8(X_{LR}),\ F_{12}(X_{ILR})] + B_{13}) \tag{5}$$

Note that the activation function (ReLU) is not applied to the last feature map when the convolution operation is conducted in Conv15. Table 1 presents the structural analysis of the network parameters in SR-ILLNN.

Table 1. Analysis of network parameters in SR-ILLNN.

Layer Name	Kernel Size	Num. of Kernels	Padding Size	Output Feature Map (W × H × C)	Num. of Parameters
Conv1	3 × 3 × 1	64	1	N × M × 64	640
Conv2	3 × 3 × 64	64	1	N × M × 64	36,928
Conv3	3 × 3 × 128	64	1	N × M × 64	73,792
Conv4	3 × 3 × 192	64	1	N × M × 64	110,656
Conv5	1 × 1 × 64	32	0	N × M × 32	2080
Deconv1	4 × 4 × 32	32	1	2N × 2M × 32	16,416
Conv6, 7	3 × 3 × 32	32	1	2N × 2M × 32	9248
Deconv2	4 × 4 × 32	32	1	4N × 4M × 32	16,416
Conv8	3 × 3 × 32	16	1	4N × 4M × 16	4624
Conv9	5 × 5 × 1	64	2	4N × 4M × 64	1664
Conv10	3 × 3 × 64	64	1	4N × 4M × 64	36,928
Conv11	3 × 3 × 128	64	1	4N × 4M × 64	73,792
Conv12	3 × 3 × 192	16	1	4N × 4M × 16	27,664
Conv13, 14	3 × 3 × 32	32	1	4N × 4M × 32	9248
Conv15	5 × 5 × 32	1	2	4N × 4M × 1	801

3.2. Architecture of SR-SLNN

Because SR-ILLNN has hierarchical network structure due to the two inputs, we propose a SR-SLNN to reduce the network complexity of SR-ILLNN. As depicted in Figure 3, the SR-SLNN was modified to remove the HR feature layers and the shared feature layers of SR-ILLNN. In addition, it has seven convolution layers and two deconvolution layers, where two convolution layers between deconvolution layers are also removed. Table 2 presents the structural analysis of network parameters in SR-SLNN.

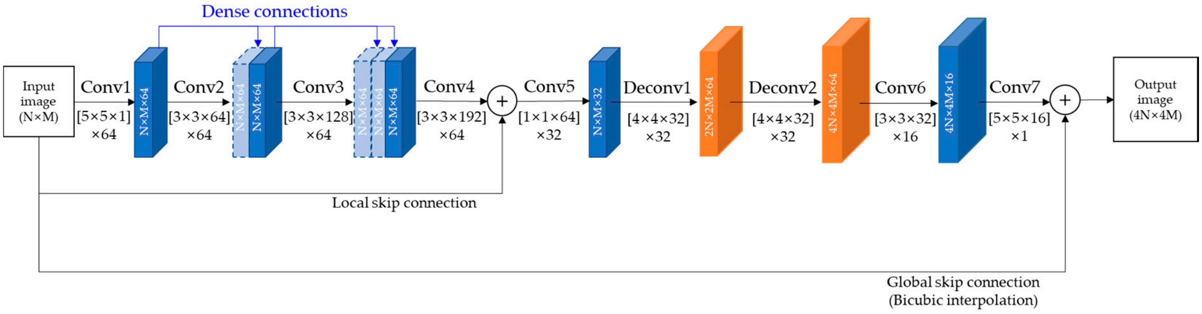


Figure 3. The framework of the proposed SR-SLNN.

Table 2. Analysis of network parameters in SR-SLNN.

Layer Name	Kernel Size	Num. of Kernels	Padding Size	Output Feature Map (W × H × C)	Num. of Parameters
Conv1	5 × 5 × 1	64	2	N × M × 64	1664
Conv2	3 × 3 × 64	64	1	N × M × 64	36,928
Conv3	3 × 3 × 128	64	1	N × M × 64	73,792
Conv4	3 × 3 × 192	64	1	N × M × 64	110,656
Conv5	1 × 1 × 64	32	0	N × M × 32	2080
Deconv1	4 × 4 × 32	32	1	2N × 2M × 32	16,416
Deconv2	4 × 4 × 32	32	1	4N × 4M × 32	16,416
Conv6	3 × 3 × 32	16	1	4N × 4M × 16	4624
Conv7	5 × 5 × 16	1	2	4N × 4M × 1	401

3.3. *Loss Function and Hyper-Parameters*

We set hyper-parameters as presented in Table 3. In order to find the optimal parameter θ = {Filter weights, Biases}, we defined mean square error (MSE) as the loss function (6), where X_{HR}, Y, and N are the final output image of SR-LNN, the corresponding label image, and the batch size. Here, $L(\theta)$ is minimized by Adam optimizer using the back-propagation. In particular, the number of epochs were set to 50 according to the Peak Signal-to-Nosie Ratio (PSNR) variations of the validation set (Set5) and the learning rates were empirically assigned to the intervals of epoch.

Since it is important to set the optimal number of network parameters in the design of lightweight neural network, we investigated the relation between the number of parameters and PSNR according to the various filter sizes. As measured in Table 4, we implemented the most of convolution layers with 3x3 filter size, except for deconvolution layers to generate the interpolated feature map that accurately corresponds to the scaling factor:

$$L(\theta) \ = \frac{1}{N}\sum_{i=0}^{N-1} \|X_{HR}\left(X^i\right) - Y^i\|_2^2 \tag{6}$$

Table 3. Hyper-parameters of the proposed methods.

Optimizer	Adam
Learning Rate	10^{-3} to 10^{-5}
Activation function	ReLU
Padding Mode	Partial convolutional based padding [23]
Num. of epochs	50
Batch size	128
Initial weight	Xavier

Table 4. Relation between the number of parameters and PSNR according to the various filter sizes.

Networks	Filter Size (Width × Height)	Num. of Parameters	PSNR (dB)
SR-ILLNN	3 × 3	439,393	31.41
	5 × 5	1,153,121	31.38
	7 × 7	2,223,713	31.29
	9 × 9	3,651,169	28.44
SR-SLNN	3 × 3	262,977	31.29
	5 × 5	664,385	31.19
	7 × 7	1,266,497	31.16
	9 × 9	2,069,313	31.15

4. Experimental Results

As shown in Figure 4, we used the DIVerse 2K (DIV2K) dataset [24] whose total number is 800 images to train the proposed methods. In order to design SR-LNN capable of up-sampling input images four times, all training images with RGB components are converted into YUV components and extracted only Y component with the size of 100 × 100 patch without overlap. In order to generate interpolated input images, the patches are down-sampled and then up-sampled it again by bi-cubic interpolation.

Finally, we obtained three training datasets from DIV2K where the total number of each training dataset is 210,048 images for original images, low-resolution images, and interpolated low-resolution images, respectively. For testing our SR-LNN models, we used Set5, Set14, Berkeley Segmentation Dataset 100 (BSD100), and Urban100 as depicted in Figure 5, which are representatively used as testing datasets in most SR studies. For reference, Set5 was also used as a validation dataset.

All experiments were run on an Intel Xeon Skylake (eight cores @ 2.59 GHz) having 128 GB RAM and two NVIDIA Tesla V100 GPUs under the experimental environment described in Table 5. After setting a bicubic interpolation method as an anchor for performance comparison, we compared the proposed two SR-LNN models with SR-CNN [9], AR-CNN [14], and SR-DenseNet [18] in terms of image quality enhancement and network complexity.

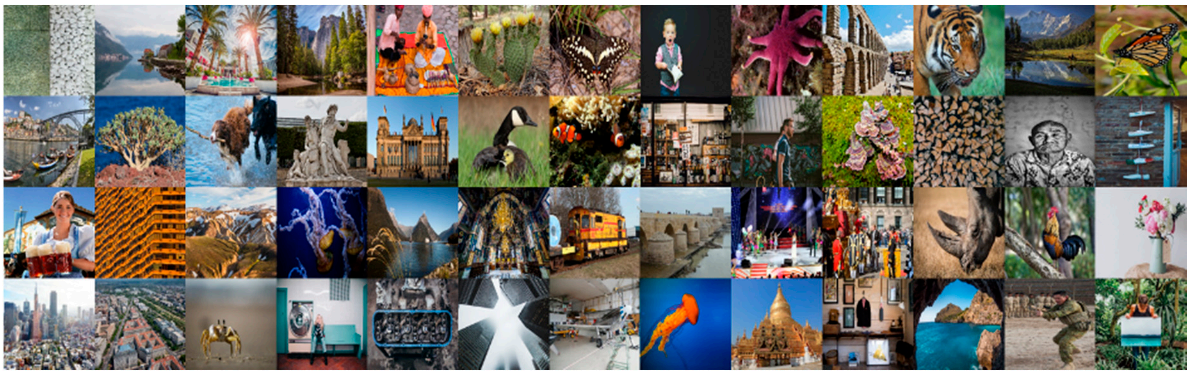


Figure 4. Training dataset. (DIV2K).

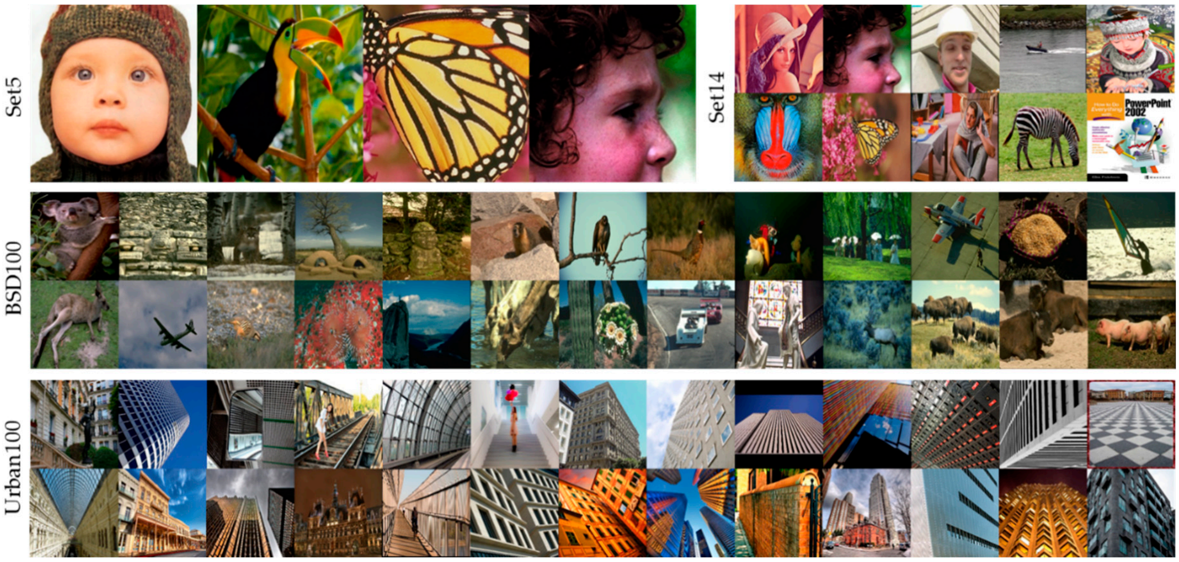


Figure 5. Test datasets. (Set5, Set14, BSD100, and Urban100).

In order to evaluate the accuracy of SR, we used PSNR and the structural similarity index measure (SSIM) [25,26] on the Y component as shown in Tables 6 and 7, respectively. In general, PSNR has been commonly used as a fidelity measurement and it is the ratio between the maximum possible power of an original signal and the power of corrupting noise that affects the fidelity of its representation. In addition, SSIM is a measurement that calculates a score using structural information of images and is evaluated as similar to human perceptual scores. Compared with the anchor, the proposed SR-ILLNN and SR-SLNN enhance PSNR by as many as 1.81 decibel (dB) and 1.71 dB, respectively. Similarly, the proposed SR-LNNs show significant PSNR enhancement, compared to SR-CNN and AR-CNN. In contrast to the results of the anchor, the proposed SR-ILLNN has similar PSNR performance on most test datasets, compared with SR-DenseNet.

Table 5. Experimental environments.

Num. of Training Samples	210,048
Input size (X_{LR})	25 × 25 × 1
Interpolated input size (X_{ILR})	100 × 100 × 1
Label size (Y_{HR})	100 × 100 × 1
Linux version	Ubuntu 16.04
CUDA version	10.1
Deep learning frameworks	Pytorch 1.4.0

Table 6. Average results of PSNR (dB) on the test dataset.

Dataset	Bicubic	SR-CNN [9]	AR-CNN [14]	SR-DenseNet [18]	SR-ILLNN	SR-SLNN
Set5	28.44	30.30	30.35	31.43	31.41	31.29
Set14	25.80	27.09	27.10	27.84	27.83	27.73
BSD100	25.99	26.86	26.86	27.34	27.33	27.28
Urban100	23.14	24.33	24.34	25.30	25.32	25.18
Average	24.73	25.80	25.81	26.53	26.54	26.44

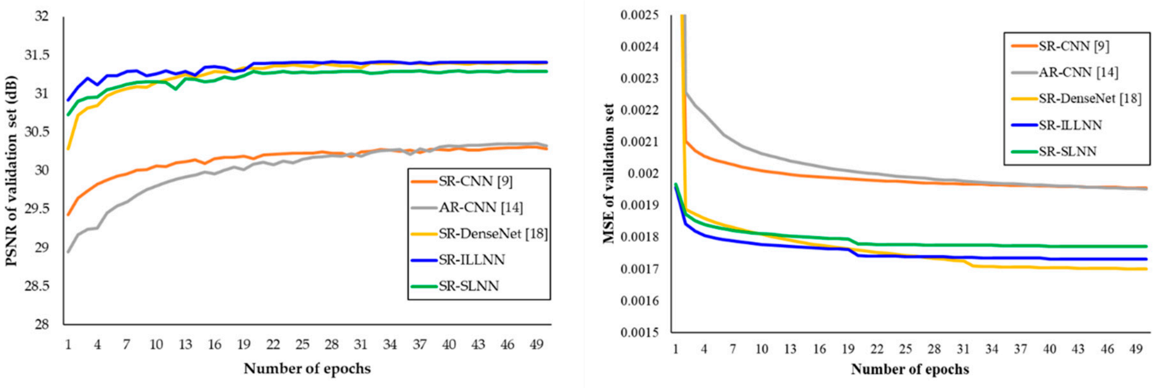
Table 7. Average results of SSIM on the test dataset.

Dataset	Bicubic	SR-CNN [9]	AR-CNN [14]	SR-DenseNet [18]	SR-ILLNN	SR-SLNN
Set5	0.8112	0.8599	0.8614	0.8844	0.8848	0.8827
Set14	0.7033	0.7495	0.7511	0.7708	0.7709	0.7689
BSD100	0.6699	0.7112	0.7126	0.7279	0.7275	0.7260
Urban100	0.6589	0.7158	0.7177	0.7584	0.7583	0.7532
Average	0.6702	0.7192	0.7208	0.7481	0.7479	0.7447

In addition, we conducted an experiment to verify the effectiveness of skip connections and dense connections. In particular, the more dense connections are deployed in the between convolution layers, the more network parameters are required in the process of convolution operations. Table 8 shows the results of tool-off tests on the proposed methods. As both skip connections and dense connections contribute to improve PSNR in the test datasets, the proposed methods are deployed these schemes. Figure 6 shows MSE as well as PSNR corresponding to the number of epochs and these experiments were evaluated from all comparison methods (SR-CNN, AR-CNN, and SR-DenseNet), including the proposed methods. It is confirmed that although SR-DenseNet has the highest reduction-rate in terms of MSE, the proposed methods have an almost similar increase rate in terms of PSNR. Figure 7 shows the comparisons of subjective quality between the proposed methods and previous methods.

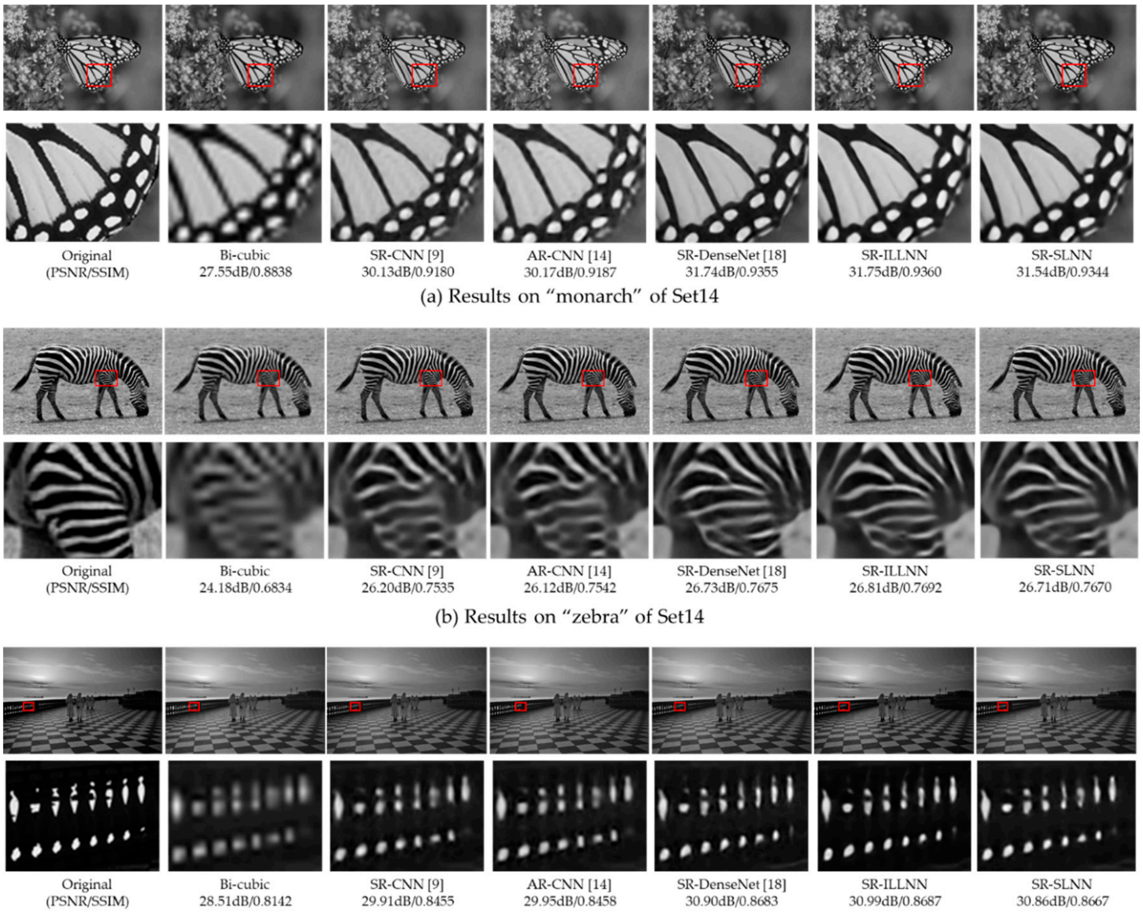
Table 8. The results of tool-off tests.

Skip Connections	Dense Connections	Set5 (PSNR)		Set14 (PSNR)		BSD100 (PSNR)		Urban100 (PSNR)	
		SR-ILLNN	SR-SLNN	SR-ILLNN	SR-SLNN	SR-ILLNN	SR-SLNN	SR-ILLNN	SR-SLNN
Disable	Disable	31.34	31.15	27.80	27.62	27.31	27.20	25.27	25.01
Enable	Disable	31.35	31.21	27.81	27.68	27.33	27.26	25.31	25.13
Disable	Enable	31.40	31.18	27.81	27.65	27.32	27.23	25.29	25.07
Enable	Enable	31.41	31.29	27.83	27.73	27.33	27.28	25.32	25.18



(a) **(b)**

Figure 6. PSNR and MSE corresponding to the number of epochs. (**a**) PSNR per epoch. (**b**) MSE per epoch.



(a) Results on "monarch" of Set14

(b) Results on "zebra" of Set14

(c) Results on "img028" of Urban100

Figure 7. Comparisons of subjective quality on test dataset. (**a**) Results on "monarch" of Set14. (**b**) Results on "zebra" of Set14. (**c**) Results on "img028" of Urban100.

In terms of the network complexity, we analyzed the number of parameters, parameter size (MB), and total memory size (MB) where total memory size includes intermediate feature maps as well as the parameter size. In general, both the total memory size and the inference speed are proportional to the number of parameters. Table 9 presents the number of parameters and total memory size. Compared with SR-DenseNet, the proposed two SR-LNNs reduce the number of parameters by as low as 8.1% and 4.8%, respectively. Similarly, the proposed two SR-LNNs reduce total memory size by as low as 35.9% and 16.1%, respectively. In addition, we evaluated the inference speed on BSD100 test images. As shown in Figure 8, the inference speed of the proposed methods is much faster than that of SR-DenseNet. Even though the proposed SR-SLNN is slower than SR-CNN and AR-CNN, it is obviously superior to SR-CNN and AR-CNN in terms of PSNR improvements as measured in Tables 6 and 7.

Table 9. Analysis of the number of parameters and memory size.

	SR-CNN [9]	**AR-CNN [14]**	**SR-DenseNet [18]**	**SR-ILLNN**	**SR-SLNN**
Num. of parameters	57,281	106,564	5,452,449	439,393	262,977
Parameter size (MB)	0.22	0.41	20.80	1.68	1.00
Total memory size (MB)	14.98	17.61	224.81	80.83	36.21

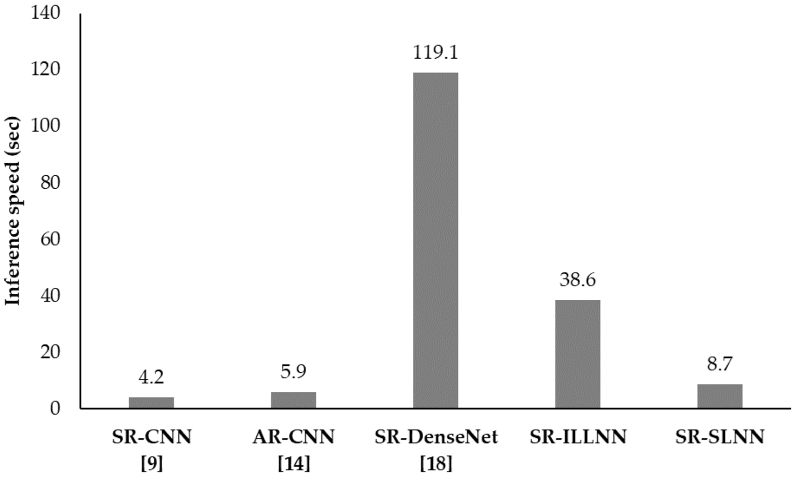


Figure 8. Inference speed on BSD100.

5. Conclusions and Future Work

In this paper, we have proposed two SR-based lightweight neural networks (SR-ILLNN and SR-SLNN) for single image super-resolution. We investigated the trade-offs between the accuracy of SR (PSNR and SSIM) and the network complexity, such as the number of parameters, memory capacity, and inference speed. Firstly, SR-ILLNN was trained on both low-resolution and high-resolution images. Secondly, SR-SLNN was designed to reduce the network complexity of SR-ILLNN. For training the proposed SR-LNNs, we used the DIV2K image dataset and evaluated both the accuracy of SR and the network complexity on Set5, Set14, BSD100, and Urban100 test image datasets. Our experimental results show that the SR-ILLNN and SR-SLNN can significantly reduce the number of parameters by 8.1% and 4.8%, respectively, while maintaining similar image quality compared to the previous methods. As future work, we plan to extend the proposed SR-LNNs to other color components as well as a luminance component for improving SR performance on color images.

Author Contributions: Conceptualization, S.K. and D.J.; methodology, S.K. and D.J.; software, S.K.; validation, D.J., B.-G.K., H.L., and E.R.; formal analysis, S.K. and D.J.; investigation, S.K. and D.J; resources, D.J.; data curation, S.K.; writing—original draft preparation, S.K.; Writing—Review and editing, D.J.; visualization, S.K.; supervision, D.J.; project administration, D.J.; funding acquisition, H.L. and E.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Ministry of Science and ICT(MSIT) of the Korea government, grant number20PQWO-B153369-02.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: This research was supported by a grant (20PQWO-B153369-02) from Smart road lighting platform development and empirical study on testbed Program funded by Ministry of Science and ICT(MSIT) of the Korea government. This research was results of a study on the "HPC Support" Project, supported by the 'Ministry of Science and ICT' and NIPA.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Yang, W.; Zhang, X.; Tian, Y.; Wang, W.; Xue, J.; Liao, Q. Deep Learning for Single Image Super-Resolution: A Brief Review. *IEEE Trans. Multimed.* **2019**, *21*, 3106–3121. [CrossRef]
2. Anwar, S.; Khan, S.; Barnes, N. A deep journey into super-resolution: A survey. *arXiv* **2019**, arXiv:1904.07523. [CrossRef]
3. Wang, Z.; Chen, J.; Hoi, S.C.H. Deep Learning for Image Super-resolution: A Survey. *IEEE Trans. Pattern. Anal. Mach. Intell.* **2020**. [CrossRef] [PubMed]
4. Keys, R. Cubic convolution interpolation for digital image processing. *IEEE Trans. Acoust.* **1981**, *29*, 1153–1160. [CrossRef]
5. Freeman, T.; Jones, R.; Pasztor, C. Example-based super-resolution. *IEEE Comput. Graph. Appl.* **2002**, *22*, 56–65. [CrossRef]
6. Kim, J.; Kim, B.; Roy, P.; Partha, P.; Jeong, D. Efficient facial expression recognition algorithm based on hierarchical deep neural network structure. *IEEE Access* **2019**, *7*, 41273–41285. [CrossRef]
7. Kim, J.; Hong, G.; Kim, B.; Dogra, D. deepGesture: Deep learning-based gesture recognition scheme using motion sensors. *Displays* **2018**, *55*, 38–45. [CrossRef]
8. Jeong, D.; Kim, B.; Dong, S. Deep Joint Spatiotemporal Network (DJSTN) for Efficient Facial Expression Recognition. *Sensors* **2020**, *20*, 1936. [CrossRef] [PubMed]
9. Dong, C.; Loy, C.C.; He, K.; Tang, X. Image super-resolution using deep convolutional networks. *IEEE Trans. Pattern. Anal. Mach. Intell.* **2015**, *38*, 295–307. [CrossRef] [PubMed]
10. Simonyan, K.; Zisserman, A. Very deep convolution networks for large -scale image recognition. In Proceedings of the International Conference on Learning Representations, San Diego, CA, USA, 7–9 May 2015.
11. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the Conference on Computer Vision and Pattern Recognition, Las Vegas, NY, USA, 27–30 June 2016; pp. 770–778.
12. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.
13. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Proceedings of the Machine Learning Research, Lille, France, 7–9 July 2015; pp. 448–456.
14. Dong, C.; Deng, Y.; Change Loy, C.; Tang, X. Compression artifacts reduction by a deep convolutional network. In Proceedings of the International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 576–584.
15. Kim, J.; Lee, J.; Lee, K. Accurate image super-resolution using very deep convolutional networks. In Proceedings of the Conference on Computer Vision and Pattern Recognition, Las Vegas, NY, USA, 27–30 June 2016; pp. 1646–1654.
16. Ledig, C.; Theis, L.; Huszár, F.; Caballero, J.; Cunningham, A.; Acosta, A.; Aitken, A.; Tejani, A.; Totz, J.; Wang, Z.; et al. Photo-realistic single image super-resolution using a generative adversarial network. In Proceedings of the Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4681–4690.
17. Lim, B.; Son, S.; Kim, H.; Nah, S.; Lee, K. Enhanced deep residual networks for single image super-resolution. In Proceedings of the Conference on Computer Vision and Pattern Recognition Workshops, Honolulu, HI, USA, 21–26 July 2017; pp. 136–144.
18. Tong, T.; Li, G.; Liu, X.; Gao, Q. Image super-resolution using dense skip connections. In Proceedings of the Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4799–4807.
19. Ye, Y.; Alshina, E.; Chen, J.; Liu, S.; Pfaff, J.; Wang, S. *[DNNVC] AhG on Deep neural networks based video coding, Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC29, Document JVET-T0121*, Input Document to JVET Teleconference. 2020.
20. Cho, S.; Lee, J.; Kim, J.; Kim, Y.; Kim, D.; Chung, J.; Jung, S. Low Bit-rate Image Compression based on Post-processing with Grouped Residual Dense Network. In Proceedings of the Conference on Computer Vision and Pattern Recognition Workshops, Long Beach, CA, USA, 16–20 June 2019.
21. Yang, R.; Xu, M.; Liu, T.; Wang, Z.; Guan, Z. Enhancing quality for HEVC compressed videos. *IEEE Trans. Circuits Syst. Video Technol.* **2018**, *29*, 2039–2054. [CrossRef]

22. Zhang, S.; Fan, Z.; Ling, N.; Jiang, M. Recursive Residual Convolutional Neural Network-Based In-Loop Filtering for Intra Frames. *IEEE Trans. Circuits Syst. Video Technol.* **2019**, *30*, 1888–1900. [CrossRef]
23. Lui, G.; Shih, K.; Wang, T.; Reda, F.; Sapra, K.; Yu, Z.; Tao, A.; Catanzaro, B. Partial convolution based padding. *arXiv* **2018**, arXiv:1811.11718.
24. Agustsson, E.; Timofte, R. NTIRE 2017 Challenge on Single Image Super-Resolution: Dataset and Study. In Proceedings of the Conference on Computer Vision and Pattern Recognition Workshops, Honolulu, HI, USA, 21–26 July 2017.
25. Wang, Z.; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Processing* **2004**, *13*, 600–612. [CrossRef] [PubMed]
26. Yang, C.; Ma, C.; Yang, M. Single-image super-resolution: A benchmark. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; pp. 372–386.